

WEST Search History

DATE: Monday, October 25, 2004

Hide?	Set Name	Query	Hit Count
		<i>DB=PGPB,USPT,USOC; PLUR=YES; OP=ADJ</i>	
<input type="checkbox"/>	L13	L9 and (control\$ same class?load\$)	0
<input type="checkbox"/>	L12	L11 and (control\$ same class?load\$)	0
<input type="checkbox"/>	L11	L10 and (new class or custom\$ or selective or detect\$)	18
<input type="checkbox"/>	L10	L3 and L9	21
<input type="checkbox"/>	L9	L8 and (reloading or reloaded or reload)	71
<input type="checkbox"/>	L8	L5	582
		<i>DB=EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ</i>	
<input type="checkbox"/>	L7	L6 and (new class or custom\$ or selective or detect\$)	6
<input type="checkbox"/>	L6	L5	26
		<i>DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ</i>	
<input type="checkbox"/>	L5	L4 and dynamic	608
<input type="checkbox"/>	L4	Class near (loading or loader)	1129
		<i>DB=PGPB,USPT,USOC; PLUR=YES; OP=ADJ</i>	
<input type="checkbox"/>	L3	L2 or L1	5544
<input type="checkbox"/>	L2	(718/1 718/100 718/101 718/102 718/103 718/104 718/105).ccls.	3590
<input type="checkbox"/>	L1	(717/146 717/147 717/148 717/149 717/150 717/151 717/152 717/153 717/154 717/155 717/156 717/157 717/158 717/159 717/160 717/161 717/162 717/163 717/164 717/165 717/166 717/167).ccls.	2026



END OF SEARCH HISTORY

BEST AVAILABLE COPY

Terms used **class loader reload**

Found **58** of **143,484**

Sort results by
Display results

 [Save results to a Binder](#)
 [Search Tips](#)
☐ Open results in a new window

Try an [Advanced Search](#)
Try this search in [The ACM Guide](#)

Results 1 - 20 of 58

Result page: [1](#) [2](#) [3](#) [next](#)

Relevance scale ☐ ☐ ☐ ☐ ☐

1 [Dynamic class loading in the Java virtual machine](#)

Sheng Liang, Gilad Bracha

October 1998 **ACM SIGPLAN Notices , Proceedings of the 13th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications**, Volume 33 Issue 10

Full text available:  [pdf\(1.03 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Class loaders are a powerful mechanism for dynamically loading software components on the Java platform. They are unusual in supporting all of the following features: *laziness*, *type-safe linkage*, *user-defined extensibility*, and *multiple communicating namespaces*. We present the notion of class loaders and demonstrate some of their interesting uses. In addition, we discuss how to maintain type safety in the presence of user-defined dynamic class loading.

2 [Programming techniques: Manipulation of Java agent bytecode to add roles](#)

Giacomo Cabri, Luca Ferrari, Letizia Leonardi

June 2003 **Proceedings of the 2nd international conference on Principles and practice of programming in Java**

Full text available:  [pdf\(215.46 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)

Roles are a powerful paradigm to develop distributed applications based on agents, especially when they are in need of interacting with other entities. An agent-oriented approach requires that roles are conceived as first-class entities, and at the same time that roles are dynamically embedded into agents at runtime. In this paper we propose an approach that addresses such requirements, enabling Java agents to dynamically assume roles. We present a mechanism that modifies the agent bytecode to a ...

3 [Technical Correspondence: Unloading Java classes that contain static fields](#)

C. E. McDowell, E. A. Baldwin

January 1998 **ACM SIGPLAN Notices**, Volume 33 Issue 1

Full text available:  [pdf\(426.95 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)

In Java, the definition of a "program" is a bit fuzzy. A Java applet is essentially a Java application (i.e. program) that can be executed by a Java enabled Web browser (i.e. an OS). An applet running inside of a browser was intended to be analogous to a conventional application running under an OS, hence the netcentric "browser is your OS" model. However, as currently implemented, this analogy breaks down with regard to the system resources allocated for classes and in particular for static fields ...

Keywords: Java, class unloading, garbage collection

4 Adding type parameterization to the Java language

Ole Agesen, Stephen N. Freund, John C. Mitchell

October 1997 **ACM SIGPLAN Notices , Proceedings of the 12th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications**, Volume 32 Issue 10

Full text available:  pdf(2.16 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Although the Java programming language has achieved widespread acceptance, one feature that seems sorely missed is the ability to use type parameters (as in Ada generics, C++ templates, and ML polymorphic functions or data types) to allow a general concept to be instantiated to one or more specific types. In this paper, we propose parameterized classes and interfaces in which the type parameter may be constrained to either implement a given interface or extend a given class. This design allows t ...

5 DrJava: a lightweight pedagogic environment for Java

Eric Allen, Robert Cartwright, Brian Stoler

February 2002 **ACM SIGCSE Bulletin , Proceedings of the 33rd SIGCSE technical symposium on Computer science education**, Volume 34 Issue 1

Full text available:  pdf(451.91 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

DrJava is a pedagogic programming environment for Java that enables students to focus on designing programs, rather than learning how to use the environment. The environment provides a simple interface based on a "read-eval-print loop" that enables a programmer to develop, test, and debug Java programs in an interactive, incremental fashion. This paper gives an overview of DrJava including its pedagogic rationale, functionality, and implementation.

6 A geographically distributed framework for embedded system design and validation

Ken Hines, Gaetano Borriello

May 1998 **Proceedings of the 35th annual conference on Design automation - Volume 00**

Full text available:  pdf(285.90 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

 [Publisher Site](#)

The difficulty of embedded system co-design is increasing rapidly due to the increasing complexity of individual parts, the variety of parts available and pressure to use multiple processors to meet performance criteria. Validation tools should contain several features in order to keep up with this trend, including the ability to dynamically change detail levels, built in protection for intellectual property, and support for gradual migration of functionality from a simulation ...

Keywords: high-level synthesis, telecommunication

7 Object oriented programming techniques to replace software components on the fly in a running program

Manfred Stadel

January 1991 **ACM SIGPLAN Notices**, Volume 26 Issue 1

Full text available:  pdf(670.70 KB)

Additional Information: [full citation](#), [abstract](#), [index terms](#)


Many systems, e. g. embedded systems, process control systems, or telecommunication

switching systems cannot be stopped running to modify and extend their software. It must be possible to replace software components on the fly by new versions while the program is running. It is shown how object oriented programming techniques can help do design and implement a software system such that its components are dynamically replaceable at run time.

8 A framework for interprocedural optimization in the presence of dynamic class loading 

Vugranam C. Sreedhar, Michael Burke, Jong-Deok Choi

May 2000 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2000 conference on Programming language design and implementation**, Volume 35 Issue 5

Full text available:  [pdf\(576.50 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Dynamic class loading during program execution in the Java Programming Language is an impediment for generating code that is as efficient as code generated using static whole-program analysis and optimization. Whole-program analysis and optimization is possible for languages, such as C++, that do not allow new classes and/or methods to be loaded during program execution. One solution for performing whole-program analysis and avoiding incorrect execution after a new class is loaded is to in ...

9 Components for distributed virtual environments 

Manuel Oliveira, Jon Crowcroft, Don Brutzman, Mel Slater

December 1999 **Proceedings of the ACM symposium on Virtual reality software and technology**

Full text available:  [pdf\(293.80 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The majority of existing systems supporting Large Scale Virtual Environments (LSVE) are based on monolithic architectures, making maintenance, reusability and extensibility difficult at best. An overview of the Java Adaptive Dynamic Environment (JADE) is presented as an alternative to the traditional approach for developing a core infrastructure for VE systems. JADE consists of a light-weight cross-platform kernel with inherent capabilities for dynamic extensibility in run-time. Although th ...

Keywords: components, framework, virtual environments, virtual reality

10 Session I: Plugging Haskell in 

André Pang, Don Stewart, Sean Seefried, Manuel M. T. Chakravarty

September 2004 **Proceedings of the ACM SIGPLAN workshop on Haskell**

Full text available:  [pdf\(153.19 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Extension languages enable users to expand the functionality of an application without touching its source code. Commonly, these languages are dynamically typed languages, such as Lisp, Python, or domain-specific languages, which support runtime *plugins* via dynamic loading of components. We show that Haskell can be comfortably used as a statically typed extension language for both Haskell and foreign-language applications supported by the Haskell FFI, and that it can perform type-safe dyn ...

Keywords: dynamic loading, dynamic typing, extension languages, functional programming, plugins, staged type inference

11 Partial behavioral reflection: spatial and temporal selection of reification 

Éric Tanter, Jacques Noyé, Denis Caromel, Pierre Cointe

October 2003 **ACM SIGPLAN Notices , Proceedings of the 18th ACM SIGPLAN conference on Object-oriented programing, systems, languages, and**


Behavioral reflection is a powerful approach for adapting the behavior of running applications. In this paper we present and motivate *partial behavioral reflection*, an approach to more efficient and flexible behavioral reflection. We expose the *spatial* and *temporal* dimensions of such reflection, and propose a model of partial behavioral reflection based on the notion of *hooksets*. In the context of Java, we describe a reflective architecture offering appropriate interf ...

Keywords: aspect-oriented programming, open systems, reflection

12 Formalizing the safety of Java, the Java virtual machine, and Java card

Pieter H. Hartel, Luc Moreau

December 2001 **ACM Computing Surveys (CSUR)**, Volume 33 Issue 4

Full text available:  pdf(442.86 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We review the existing literature on Java safety, emphasizing formal approaches, and the impact of Java safety on small footprint devices such as smartcards. The conclusion is that although a lot of good work has been done, a more concerted effort is needed to build a coherent set of machine-readable formal models of the whole of Java and its implementation. This is a formidable task but we believe it is essential to build trust in Java safety, and thence to achieve ITSEC level 6 or Common Crite ...

Keywords: Common criteria, programming

13 Portable resource control in Java

Walter Binder, Jane G. Hulaas, Alex Villazón

October 2001 **ACM SIGPLAN Notices , Proceedings of the 16th ACM SIGPLAN conference on Object oriented programming, systems, languages, and applications**, Volume 36 Issue 11

Full text available:  pdf(307.08 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Preventing abusive resource consumption is indispensable for all kinds of systems that execute untrusted mobile code, such as mobile object systems, extensible web servers, and web browsers. To implement the required defense mechanisms, some support for resource control must be available: accounting and limiting the usage of physical resources like CPU and memory, and of logical resources like threads. Java is the predominant implementation language for the kind of systems envisaged here, even th ...

Keywords: Java, bytecode rewriting, micro-kernels, mobile object systems, resource control, security

14 Back to the basics: a first class chalkboard and more

Ng S. T. Chong, Masao Sakauchi

March 2000 **Proceedings of the 2000 ACM symposium on Applied computing**

Full text available:  pdf(480.68 KB) Additional Information: [full citation](#), [references](#), [index terms](#)

Keywords: distributed Java application, online authoring system, whiteboard

15 A selective, just-in-time aspect weaver

Yoshiki Sato, Shigeru Chiba, Michiaki Tatsubori

September 2003 **Proceedings of the second international conference on Generative programming and component engineering**

Full text available:  [pdf\(256.62 KB\)](#)


Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Dynamic AOP (Aspect-Oriented Programming) is receiving growing interests in both the academia and the industry. Since it allows weaving aspects with a program at runtime, it is useful for rapid prototyping and adaptive software. However, the previous implementations of dynamic AOP systems suffered from serious performance penalties. This paper presents our new efficient dynamic AOP system in Java for addressing the underlying problem. This system called Wool is a hybrid of two approaches. When a ...

16 A specification of Java loading and bytecode verification

Allen Goldberg

November 1998 **Proceedings of the 5th ACM conference on Computer and communications security**

Full text available:  [pdf\(1.15 MB\)](#)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

Keywords: Java, bytecode verification, flow analysis, formal specification

17 A scalable architecture for multi-threaded JAVA applications

M. Mrva, K. Buchenrieder, R. Kress

February 1998 **Proceedings of the conference on Design, automation and test in Europe**

Full text available:  [pdf\(117.76 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

 [Publisher Site](#)


The paper presents a scalable architecture for multi-threaded Java applications. Threads enable modeling of concurrent behavior in a more or less natural way. Thus threads give a migration path to multi-processor machines. The proposed architecture consists of multiple application-specific processing elements, each able to execute a single thread at one time. The architecture is evaluated by implementing a portable and scalable Java machine onto an FPGA board for demonstration.

Keywords: Java, application-specific, configurable, multi-threaded

18 Cluster resource management: An integrated experimental environment for distributed systems and networks

Brian White, Jay Lepreau, Leigh Stoller, Robert Ricci, Shashi Guruprasad, Mac Newbold, Mike Hibler, Chad Barb, Abhijeet Joglekar

December 2002 **ACM SIGOPS Operating Systems Review**, Volume 36 Issue S1

Full text available:  [pdf\(2.10 MB\)](#)


Additional Information: [full citation](#), [abstract](#), [references](#)

Three experimental environments traditionally support network and distributed systems research: network emulators, network simulators, and live networks. The continued use of multiple approaches highlights both the value and inadequacy of each. Netbed, a descendant of Emulab, provides an experimentation facility that integrates these approaches, allowing researchers to configure and access networks composed of emulated, simulated, and wide-area nodes and links. Netbed's primary goals are ease ...

19 Design technologies: Visualizing and querying software structures

Mariano Consens, Alberto Mendelzon, Arthur Ryman

October 1991 **Proceedings of the 1991 conference of the Centre for Advanced Studies on Collaborative research**

Full text available:  pdf(1.18 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Software engineering problems often involve large sets of objects and complex relationships among them. This report proposes that graphical visualization techniques can help engineers understand and solve a class of these problems. To illustrate this, two problems are analyzed and recast using the graphical language GraphLog. The first problem is that of simplifying dependencies among components of a system, which translates into removing cycles from a graph. The second problem is that of design ...

Keywords: 4Thought, G+, GraphLog, Prolog, graphical queries, software design, software engineering, theory-model paradigm, visual queries, visualizations

20 Zones, contracts and absorbing changes: an approach to software evolution

Huw Evans, Peter Dickman

October 1999 **ACM SIGPLAN Notices , Proceedings of the 14th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications**, Volume 34 Issue 10

Full text available:  pdf(2.46 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper describes a novel approach to managing the evolution of distributed, persistent systems at run-time. This is achieved by partitioning a system into disjoint zones, each of which can be evolved without affecting code in any other. Contracts are defined between zones, making type-level interdependencies and inter-zone communication explicit. Programmer supplied code is added to the running system, at the boundary between zones, to constrain the sco ...

Results 1 - 20 of 58

Result page: [1](#) [2](#) [3](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2004 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)

Welcome to IEEE Xplore®

- ☐ Home
- ☐ What Can I Access?
- ☐ Log-out

Tables of Contents

- ☐ Journals & Magazines
- ☐ Conference Proceedings
- ☐ Standards

Search

- ☐ By Author
- ☐ Basic
- ☐ Advanced
- ☐ CrossRef

Member Services

- ☐ Join IEEE
- ☐ Establish IEEE Web Account
- ☐ Access the IEEE Member Digital Library

IEEE Enterprise

- ☐ Access the IEEE Enterprise File Cabinet

Print Format

Your search matched **9** of **1082760** documents.A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance Descending** order.

Refine This Search:

You may refine your search by editing the current search expression or entering new one in the text box.

☐ Check to search within this result set

Results Key:

JNL = Journal or Magazine **CNF** = Conference **STD** = Standard**1 JMangler - a framework for load-time transformation of Java class f**
Kniesel, G.; Costanza, P.; Austermann, M.;

Source Code Analysis and Manipulation, 2001. Proceedings. First IEEE Interna Workshop on , 10 Nov. 2001

Pages:98 - 108

[\[Abstract\]](#) [\[PDF Full-Text \(128 KB\)\]](#) IEEE CNF**2 Secure Java class loading***Li Gong;*

Internet Computing, IEEE , Volume: 2 , Issue: 6 , Nov.-Dec. 1998

Pages:56 - 61

[\[Abstract\]](#) [\[PDF Full-Text \(64 KB\)\]](#) IEEE JNL**3 Dynamic behaviours for computer animation: the use of Java***Palmer, I.J.;*

Computer Animation '97 , 5-6 June 1997

Pages:151 - 156

[\[Abstract\]](#) [\[PDF Full-Text \(520 KB\)\]](#) IEEE CNF**4 Implementing dynamic language features in Java using dynamic code generation***Breuel, T.M.;*

Technology of Object-Oriented Languages and Systems, 2001. TOOLS 39. 39th International Conference and Exhibition on , 29 July-3 Aug. 2001

Pages:143 - 152

[\[Abstract\]](#) [\[PDF Full-Text \(480 KB\)\]](#) IEEE CNF

5 Development of soccer agents with object migration

Maeda, K.;

Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. IEEE International Conference on , Volume: 6 , 12-15 Oct. 1999
Pages:750 - 755 vol.6

[\[Abstract\]](#) [\[PDF Full-Text \(336 KB\)\]](#) IEEE CNF

6 A real-time Java system on a multithreaded Java microcontroller

Pfeffer, M.; Uhrig, S.; Ungerer, T.; Brinkschulte, U.;

Object-Oriented Real-Time Distributed Computing, 2002. (ISORC 2002).
Proceedings. Fifth IEEE International Symposium on , 29 April-1 May 2002
Pages:34 - 41

[\[Abstract\]](#) [\[PDF Full-Text \(279 KB\)\]](#) IEEE CNF

7 Jato: a compact binary file format for Java class

Sheng-De Wang; Lin, Y.;

Parallel and Distributed Systems, 2001. ICPADS 2001. Proceedings. Eighth International Conference on , 26-29 June 2001
Pages:467 - 474

[\[Abstract\]](#) [\[PDF Full-Text \(556 KB\)\]](#) IEEE CNF

8 Using Java to add "stored procedures" to databases

Ege, R.K.; Rische, N.; Jingyu Liu; Lebedev, V.;

Technology of Object-Oriented Languages and Systems, 1999. TOOLS 30. Proceedings , 1-5 Aug. 1999
Pages:322 - 331

[\[Abstract\]](#) [\[PDF Full-Text \(192 KB\)\]](#) IEEE CNF

9 Design, and implementation of a Java execution environment

Chen, F.G.; Ting-Wei Hou;

Parallel and Distributed Systems, 1998. Proceedings., 1998 International Conference on , 14-16 Dec. 1998
Pages:686 - 692

[\[Abstract\]](#) [\[PDF Full-Text \(56 KB\)\]](#) IEEE CNF











[Home](#) | [Log-out](#) | [Journals](#) | [Conference Proceedings](#) | [Standards](#) | [Search by Author](#) | [Basic Search](#) | [Advanced Search](#) | [Join IEEE](#) | [Web Account](#) |
[New this week](#) | [OPAC Linking Information](#) | [Your Feedback](#) | [Technical Support](#) | [Email Alerting](#) | [No Robots Please](#) | [Release Notes](#) | [IEEE Online Publications](#) | [Help](#) | [FAQ](#) | [Terms](#) | [Back to Top](#)

Copyright © 2004 IEEE — All rights reserved

[Web](#) | [Images](#) | [Directory](#) | [Local NEW!](#) | [News](#) | [Products](#)
YAHOO!search

[Shortcuts](#) [Advanced Search](#) [Preferences](#)

Search Results Results **1 - 10** of about **1,880** for **"class loader" reload stack** - 0.22 sec. ([About this page](#))

1. <http://search.cpan.org/src/ADAMK/Class-Autouse-1.02/lib/Class/Autouse.pm> 
 So ALWAYS run in devel mode under mod_perl.
search.cpan.org/src/ADAMK/Class-Autouse-1.02/lib/Class/Autouse.pm - 21k - [Cached](#) - [More from this site](#)
2. <http://search.cpan.org/src/ADAMK/Class-Autouse-1.03/lib/Class/Autouse.pm> 
 So ALWAYS run in devel mode under mod_perl.
search.cpan.org/src/ADAMK/Class-Autouse-1.03/lib/Class/Autouse.pm - 21k - [Cached](#) - [More from this site](#)
3. <http://www.cs.usm.maine.edu/class/cos478/opj-1.6.5/src/java/lang/ClassLoader.java> 
 ... **class loader**. */ private Vector nativeLibraries = new Vector(); /* native libraries being loaded/unloaded.
Stack ... isAbsolute; } void reload() { try { loadLibrary ...
[cs.usm.maine.edu/class/cos478/opj-1.6.5/src/java/lang/...](http://www.cs.usm.maine.edu/class/cos478/opj-1.6.5/src/java/lang/ClassLoader.java) - 59k - [Cached](#) - [More from this site](#)
4. [JRun API: Class FilterManager](#) 
 ... optimize the call **stack**. Parameters: ... **reload**(java.lang.ClassLoader loader) Reloads the current filters
 example, if the **class loader** ...
livedocs.macromedia.com/jrun/4/.../jrun/servlet/FilterManager.html - [More from this site](#)
5. [Microsoft PowerPoint - classnote16 \(PDF\)](#) 
 ... **Class loader**. **Class Loader** ... Why Class Loaders? (cont'd) • **Reload** classes ... – Accessing/manipulating
 Accessing/modifying VM registers ...
studsys.msccs.mu.edu/~iq/mscs236/fall03/Lecture/classnote16.pdf - [More from this site](#)
6. [JUnit 3.8](#) 
 JUnit 3.8.1. 08/31/2002. JUnit is a simple framework to write repeatable tests. It is an instance of the xUnit
 testing frameworks. Summary of Changes between 3.8 and 3.8.1
www.cs.wm.edu/~noonan/junit/readme.html - [More from this site](#)
7. <http://jazz.external.hp.com/src/java/opensource/JavaCI/JavaCI.doc> (MICROSOFT WORD) 
 ... The MPE CI redo **stack** is not programmatically accessible, but redo functionality is such ... We cannot
 loaded by the default system **class loader**. For this reason ...
jazz.external.hp.com/src/java/opensource/JavaCI/JavaCI.doc - [More from this site](#)
8. [Tomcat Internal API: Index](#) 
 Overview. Package. Class. Use. Tree. Deprecated. Index. Help. A. B. C. D. E. F. G. H. I. J. K. keys() - Me
 org.apache.tomcat.util.SimpleHashtable. Returns an enumeration of the keys in this hashtable. ... Adaptive
loader that loads classes from directories ... Creates a new **class loader** that will load classes from ... last
 create a parent-child relation ...
docs.mafi-trench.com/javadoc/index-all.html - [More from this site](#)
9. [JUnit 3.8](#) 
 JUnit 3.8. 08/23/2002. JUnit is a simple framework to write repeatable tests. It is an instance of the xUnit a
 testing frameworks. Summary of Changes between 3.7 and 3.8. Framework
junit.sourceforge.net/README.html - 21k - [Cached](#) - [More from this site](#)
10. [Java Programming: Java WebStart and dynamic class loading](#) 
 ... Page Options. **Reload** This Question. Java Area. Send To A Friend ... see from the following exception

the System **class loader**, or any newly created URLClassLoader ...
experts-exchange.com/Programming/... /Java/Q_20512519.html - [More from this site](#)

Results Page:

1 2 3 4 5 6 7 8 9 10 ► **Next**

[Web](#) | [Images](#) | [Directory](#) | [Local](#) ^{NEW!} | [News](#) | [Products](#)

Your Search: "class loader" reload stack

Search

Help us improve your search experience. [Send us feedback.](#)

Create your own personal search experience with [My Yahoo! Search](#) [BETA]

Copyright © 2004 Yahoo! Inc. All rights reserved. [Privacy Policy](#) - [Terms of Service](#) - [Submit Your Site](#)



[Web](#) [Images](#) [Groups](#) [News](#) [Froogle](#) [more »](#)

"class loader" reload stack

Search

[Advanced Search](#)
[Preferences](#)

Web

Results 1 - 10 of about 1,230 for "**class loader**" **reload stack**. (0.40 seconds)

Did you mean: "**classloader**" reload stack

Sponsored Links

Java Technology Forums

... in memory, so the > only way to **reload** a class ... but you are free to write your own **class loader** that does ... as they're not participating in an active call **stack**). ...

[forum.java.sun.com/thread.jsp?forum=4&](http://forum.java.sun.com/thread.jsp?forum=4&thread=435579&tstart=60&trange=15-40k)

[thread=435579&tstart=60&trange=15 - 40k](http://forum.java.sun.com/thread.jsp?forum=4&thread=435579&tstart=60&trange=15-40k) - [Cached](#) - [Similar pages](#)

Discount Stack Loader

All cameras and so much more! aff
Check out **Stack Loader** now.
www.ebay.com

[See your message here...](#)

Java Technology Forums

... this makes complete sense because the **class loader** Tomcat uses ... What is the exception specifically with a **stack** trace ... problem if you plan to unload/reload any of ...
[forum.java.sun.com/ thread.jsp?thread=481044&forum=4&message=2246216](http://forum.java.sun.com/thread.jsp?thread=481044&forum=4&message=2246216) - 64k -

[Cached](#) - [Similar pages](#)

[[More results from forum.java.sun.com](#)]

Echidna - A free multiprocess system in Java

... launch the process so you can **reload** new versions ... classes not loaded by the process' **class loader** - typically, AWT ... classloader will be on the **stack** and this ...

www.javagroup.org/echidna/ - 15k - [Cached](#) - [Similar pages](#)

Index of /~rebelsky/Courses/CS223/2004F/Examples/junit3.8.1

... warning too obtrusive [417978] constructor **stack** trace, please [415103] **Reload** checkbox should be ... a check box to enable/disable the custom **class loader**. ...

[www.cs.grinnell.edu/~rebelsky/ Courses/CS223/2004F/Examples/junit3.8.1/](http://www.cs.grinnell.edu/~rebelsky/Courses/CS223/2004F/Examples/junit3.8.1/) - 23k -

[Cached](#) - [Similar pages](#)

SourceForge.net: junit-devel

... Obviously we can't **reload** the JUnit framework ... blocking of the application **class loader** impacts security ... true; } - /* Returns a filtered **stack** trace */ @ -237 ...

[sourceforge.net/mailarchive/forum.php?forum_](http://sourceforge.net/mailarchive/forum.php?forum_id=3274&style=flat&viewday=11&viewmonth=200111)

[id=3274&style=flat&viewday=11&viewmonth=200111](http://sourceforge.net/mailarchive/forum.php?forum_id=3274&style=flat&viewday=11&viewmonth=200111) - 80k - Supplemental Result -

[Cached](#) - [Similar pages](#)

JRun API: Class FilterManager

... the servlet engine can optimize the call **stack**. ... public void **reload** (java.lang.ClassLoader loader) Reloads the ... for example, if the **class loader** changes Parameters ...

[livedocs.macromedia.com/jrun/ 4/javadocs/jrun/servlet/FilterManager.html](http://livedocs.macromedia.com/jrun/4/javadocs/jrun/servlet/FilterManager.html) - 25k -

[Cached](#) - [Similar pages](#)

[doc] JavaCI User's Guide

File Format: Microsoft Word 97 - [View as HTML](#)

... the JavaCI, both the JavaCI's redo **stack** size and the redo **stack** size of ... We cannot **reload** classes that are loaded by the default system **class loader**. ...

[jazz.external.hp.com/src/ java/opensource/JavaCI/JavaCI.doc](http://jazz.external.hp.com/src/java/opensource/JavaCI/JavaCI.doc) - [Similar pages](#)

[PDF] Enabling Mobile Agents to Dynamically Assume Roles

File Format: PDF/Adobe Acrobat - [View as HTML](#)

... system is based on a special **class loader**, called "role ... After the **reload** event the

<http://www.google.com/search?hl=en&q=%22class+loader%22+reload+stack&b...> 10/25/04

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.